



**TODO1 ruby 纯面向对象的语言，**

所操作的一切都是对象，操作的结果也是对象。每个值都是一个对象，即使是最原始的东西：字符串、数字，甚至连 true 和 false 都是对象，类本身也是一个对象。

比如 取绝对值

* java int num = Math.abs(-5)；把一个数值传递给 Math 类的一个静态函数 abs 处理(数值是基本类型,不是类)
* Ruby c = -5.abs (数字–99就是对象，取绝对值的操作属于数字本身)

**TODO2 java7-java8 语法不同**

1. lambda表达式　ラムダ式（闭包）

---需要一个函数，但又不想费神去命名一个函数的场合下使用，也就是指匿名函数--- 允许把函数作为一个方法的参数（函数作为参数传递进方法中）

2. 接口可以添加默认方法，default;

在JAVA7及之前的版本对接口的要求：

接口定义：

* 使用 interface 关键字 。
* 接口中的所有 成员变量都默认是由 public static final 修饰的。
* 接口中的所有方法都默认是由 public abstract 修饰的。也可以使用 protected ，但不能用 private 。

接口中的 所有方法都没有方法体。

接口没有构造方法。因为构造方法用于创建对象。

实现接口的类 必须提供接口中所有方法的具体实现内容。

可声明 静态常量、属性和 抽象方法：

全局变量（ public static final ）和 抽像方法（ public abstract method() )

**Java8**允许给接口添加一个非抽象方法 实现，只需要使用 default关键字即可

在 JDK7 的基础上可以声明 静态方法 和 默认方法，必须写方法体

默认方法用于接口的扩展，实现类可用，可重写。

静态方法用于接口的扩展，接口类名直接调用，实现类不可调用，不可重写。

1. spring生态各种版本的支持不同，比如 springboot，8 支持，7 不支持

**封装(カプセル化) 的概念 数据隐藏**

封装把一个对象的属性私有化，同时提供一些可以被外界访问的属性的方法

java中通过关键字private，protected和public实现封装，Encap

* 将类的变量声明为private。
* 提供公共setter和getter方法来修改和查看变量值。

封装属性：

* private 属性类型 属性名

封装方法：

* private 方法返回类型 方法名称（参数）